

Taming Diffusion Transformer for Real-Time Mobile Video Generation

Yushu Wu^{1,2*,†} Yanyu Li^{1†} Anil Kag¹ Ivan Skorokhodov¹
 Willi Menapace¹ Ke Ma¹ Arpit Sahni¹ Ju Hu¹ Aliaksandr Siarohin¹
 Dhritiman Sagar¹ Yanzhi Wang² Sergey Tulyakov¹
¹Snap Inc. ²Northeastern University
 Project Page: https://snap-research.github.io/mobile_video_dit/



Figure 1: Videos generated by our efficient Diffusion Transformer.

Abstract

Diffusion Transformers (DiT) have shown strong performance in video generation tasks, but their high computational cost makes them impractical for resource-constrained devices like smartphones, and real-time generation is even more challenging. In this work, we propose a series of novel optimizations to significantly accelerate video generation and enable real-time performance on mobile platforms. First, we employ a highly compressed variational autoencoder (VAE) to reduce the dimensionality of the input data without sacrificing visual quality. Second, we introduce a KD-guided, sensitivity-aware tri-level pruning strategy to shrink the model size to suit mobile platform while preserving critical performance characteristics. Third, we develop an adversarial step distillation technique tailored for DiT, which allows us to reduce the number of inference steps to four. Combined, these optimizations enable our model to achieve over 10 frames per second (FPS) generation on an iPhone 16 Pro Max, demonstrating the feasibility of real-time, high-quality video generation on mobile devices.

1 Introduction

The rapid advancement of generative models [25, 6, 33] has led to significant breakthroughs in video generation [3, 29, 59, 44, 53, 42, 18], with Diffusion Transformers emerging as one of the most effective architectures for producing temporally coherent and visually compelling video content. These models leverage the strengths of diffusion processes [33, 12, 25] for stepwise refinement and transformer-based attention for capturing long-range dependencies across frames, making them particularly suitable for generating complex, high-fidelity video sequences. As such, they have become a cornerstone in state-of-the-art video synthesis pipelines.

*Work done during an internship at Snap Inc.

†Equal contributions

Despite their impressive generative capabilities, Diffusion Transformers suffer from substantial computational overhead, especially when applied to high-resolution video generation. Though delivering a great quality boost, the computation and memory consumption of the 3D full attention [44, 53, 18, 9] scale quadratically with respect to total tokens

($t \times H \times W$). This limitation poses a critical challenge for deploying these models in real-time or interactive settings, particularly on mobile devices with limited processing power and energy budgets. Existing efforts to optimize diffusion-based models, such as step reduction [57, 21, 54], efficient backbone [49, 51, 58], are mainly focused on Unet-based denoisers, which are naturally less expressive. Very few work [9] investigates the efficiency of DiT, and often suffers from perceptual quality or temporal consistency loss. Furthermore, most current acceleration methods are designed for server-class hardware [9, 44] and do not translate well to edge devices.

In this work, we present a comprehensive optimization pipeline tailored specifically to accelerate video diffusion transformers for mobile deployment. Our approach combines three key strategies.

(A) High-Compression Video Variational Autoencoder (VAE). First, we investigate the compression rate of the VAE. Video VAE with a higher compression ratio can significantly reduce the number of tokens in the latent representation, thus speeding up DiT inference. However, VAEs with an aggressive compression ratio often suffer from a loss of reconstruction quality, which likely leads to a loss in diffusion model generation quality. The trade-off between compression ratio and diffusion quality remains underexplored for on-device video generation. In this work, we create a series of video VAEs with different compression ratios and compare the speed gain versus generation quality loss. We have several findings: (i) the reconstruction and diffusion generation quality corresponds well with the compression ratio. (ii) The speedup from the higher VAE compression ratio is significant. (iii) Though slightly degraded, we can still find a sweet point that balances speed and quality.

(B) Efficient Mobile DiT. Second, we find that directly training a lightweight DiT designated for mobile is challenging. Instead, we start from a larger pre-trained supernet and propose a sensitivity aware tri-level pruning with KD-Guided framework that selectively removes less critical components of the model based on their contribution to both runtime and output quality. This pruning reduces the number of DiT blocks, feed-forward features, and attention heads. The final architecture has 915M parameters and can be easily deployed on a modern device such as an iPhone 16 Pro Max. Further, we improve the pruned model performance by aligning features of the pruned network and the supernet through knowledge distillation.

(C) Adversarial Step Distillation. Finally, we introduce a novel adversarial distillation method for DiTs. It allows us to achieve comparable quality to full-step diffusion model using only a small number of inference steps, dramatically reducing computational cost. Prior adversarial distillation methods for video diffusion models mainly focus on Unet backbones [51, 57] and less challenging image-to-video tasks [3]. We find that they cannot be readily applied to diffusion transformers. We design a new discriminator architecture where we inherit the first K blocks from the generator and freeze them as the time-conditioned feature parser, and add both full 3D attention and cross attention as learnable discriminator layers. Our design enables 4 step generation without classifier-free guidance (CFG), which is $20\times$ faster compared with a typical 40-step recipe with CFG.

With these optimizations, our model can generate high-quality video at over 10 frames per second (FPS) on an iPhone 16 Pro Max using only four denoising steps. Extensive experiments demonstrate that our method maintains strong visual fidelity and temporal consistency, closely matching the outputs of full-resolution, unpruned models. For the first time, our work advances the state-of-the-art for on-device efficient video generation by making real-time diffusion-based video synthesis feasible on consumer-grade mobile hardware. Our contributions can be summarized as follows,

- We are the first to systematically investigate the trade-off between latent compression ratio, generation quality, and speed for on-device video generation. We find that for diffusion transformer, a $8 \times 32 \times 32$ VAE achieves a good trade-off between generation speed and quality.
- To obtain an efficient DiT backbone, we find that training a smaller network from scratch yields inferior generation results. We instead start from a large pre-trained super network and perform

Model	Params (B)	VBench	A100	iPhone
Wan2.1	1.3	83.33	0.2	\times
LTX	1.8	80.00	6.1	\times
Ours-Server	2.0	83.09	6.4	\times
Ours-Mobile	0.9	81.45	151.3	12.2

Table 1: Our model is the first DiT-based mobile video generator. Speed are converted to generation FPS (i.e. generating 121 frames in 12s is 10FPS). Details in Sec. 5.

distillation-guided, sensitivity-aware pruning. We deliver a compact network with optimized depth and width configuration.

- For adversarial step-distillation, we propose a new discriminator design tailored for DiTs which outperforms prior methods by a large margin. We achieve 4-step inference without CFG.

2 Related Work

Video Diffusion Model There has been an exponential growth in the literature for video generation models [44, 29, 24, 53, 18, 42, 19, 26]. Most of the recent efforts have been towards developing large video diffusion models, which learn to iteratively denoise a pure Gaussian noise into a semantically meaningful and content-rich video sample conditioned on user inputs such as text-prompt or image. These include pixel space [28, 11] and latent space [44, 53] video models. While these models [29, 59, 28, 32, 9, 53] are capable of generating realistic and visually appealing videos, their resource requirements render them unsuitable for on-device deployment.

There have been very little efforts towards on-device video generation [49, 17]. Wan2.1 [44] family includes a 1.3B T2V model, but due to low VAE compression ratio, it yields large number of latent tokens prohibiting the deployment on-device. LTX-Video [10] model uses a high-compression VAE, resulting in compressed latents. Although LTX-Video generates videos in real-time and is considerably faster than Wan2.1, their model size (1.9B) is still too restrictive to be deployed on-device. SnapGen-V [49] utilizes a tiny UNet denoiser for diffusion modeling, but the generated videos have poor quality. Mobile Video Diffusion [51] prunes the Stable Video Diffusion [3] model and reduces the number of channels and UNet blocks. On-device Sora [17] enables low resolution video generation on iPhones by enabling token merging in temporal dimension and concurrent model block loading to address the limited working memory. In this work, we propose an efficient mobile DiT architecture and utilize a high-compression VAE to design an on-device real-time video generation model which generates high-quality high-resolution videos on mobile devices such as iPhone 16 Pro.

Step Distillation. Since diffusion models [6, 31, 13] iteratively denoise the gaussian noise, they require substantial number of steps, each requiring a forward pass through a large neural network, to generate a video. Thus, reducing the number of inference steps directly cuts down the video generation latency. There have many works [55, 54, 52, 47, 40, 39, 57, 49] that aims to solve this problem of step distillation in diffusion models. Most of these works have been for the text-to-image models [55, 54, 52, 47, 16, 27, 5].

Early works [34, 22] proposed progressive distillation wherein a student model learns to predict the output of two inference steps of a pretrained teacher. This procedure when repeated yields a few step diffusion model, but this requires progressively refining the inference steps via a new training procedure, resulting in computationally expensive step distillation approach. Consistency Models [40, 39] refine the distillation objective to enforce a consistency property where each point in trajectory maps to the clean data. Later works [50, 38, 35] further incorporate adversarial loss to distill a single-step student, and enhance multi-step results as well. Recently proposed Shortcut models [8] introduce flow-matching loss with self-consistency loss while modifying the denoiser to be aware of the inference step.

In the context of video models, there have been works such as SF-V [57] and SnapGen-V [49] that enables few step video generation through step distillation. SF-V [57] enables one-step video generation for the SVD [3] model by adding latent adversarial training. Inspired by SF-V, SnapGen-V [49] proposes a few step video generation model for an efficient UNet denoiser by developing a unified spatio-temporal discriminator design in the adversarial training.

3 Preliminaries

We recap the basics of video diffusion models and define our annotations. Following popular practices of latent diffusion [59], we employ a video autoencoder to encode video data $\mathbf{X} \in \mathbb{R}^{3 \times T \times H \times W}$ into a compressed latent space $\mathbf{x} \in \mathbb{R}^{c \times t \times h \times w}$, where T is the number of temporal frames, H and W are the spatial resolutions, and c is the latent channels. The VAE compression ratio is thus $\frac{T}{t} \times \frac{H}{h} \times \frac{W}{w}$, e.g., $4 \times 8 \times 8$ [53, 18, 44] and $8 \times 16 \times 16$ [2] VAEs. The objective of the DiT generator is to generate \mathbf{x} under certain guidance (i.e., text prompt).

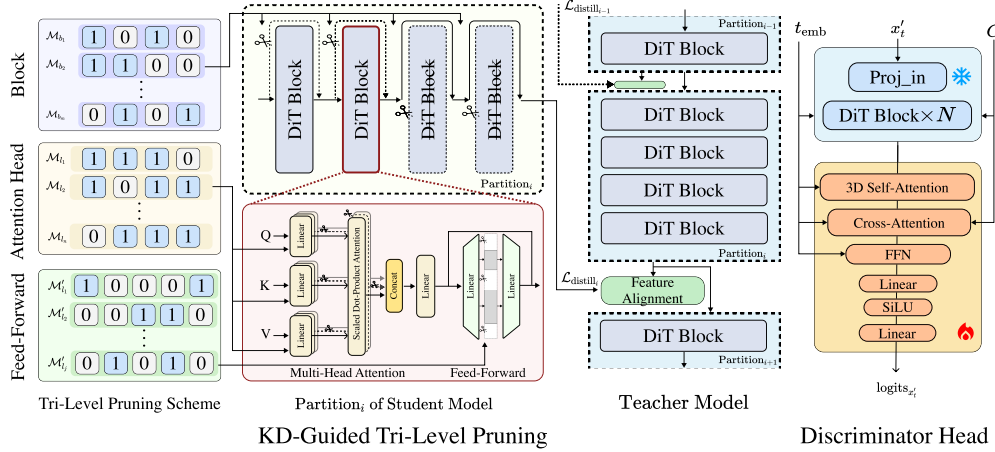


Figure 2: **Overview of proposed KD-Guided Tri-Level Pruning and new discriminator head.** The tri-level pruning scheme operates across three levels of granularity, the block, attention-head, and feed-forward network dimension, ranging from coarse to fine. This design enables flexible, efficient, and stable model compression. Additionally, the proposed discriminator adopts standard diffusion transformer block components with a MLP classifier head, improved condition alignment for adversarial training.

We employ Rectified Flow [47] to train our latent DiT model. According to the flow-matching-based diffusion process, given a clean video latent $\mathbf{x}_0 = \mathbf{x}$, the intermediate noisy state \mathbf{x}_t at a timestep t is:

$$\mathbf{x}_t = (1 - t) \mathbf{x}_0 + t\epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, I), \quad (1)$$

which is a linear interpolation between the data distribution and a standard normal distribution. The model aims to learn a vector field $v_\theta(t, \mathbf{x}_t)$ using the Conditional Flow Matching objective, *i.e.*,

$$\mathcal{L}_{\text{flow-matching}} = \mathbb{E}_{t, \epsilon, \mathbf{x}_0} \|v_\theta(t, \mathbf{x}_t) - (\epsilon - \mathbf{x}_0)\|_2^2. \quad (2)$$

4 Method

4.1 Scaling Latent Compression Ratio

DiT demonstrates superior generation capabilities when attending on full token length (*thw*) [53], however, it is also notorious for its quadratic computational cost. The key idea of the latent diffusion model is to construct a compressed latent space and reduce the generation cost. As a result, a straightforward idea to accelerate DiT is to further increase the VAE compression ratio. State-of-the-art models (CogVideoX [53], Hunyuan[18], Wan [44]) employ a $4 \times 8 \times 8$ VAE combined with a $1 \times 2 \times 2$ patchify module, which comprises a $4 \times 16 \times 16$ total compression rate, while the recent OpenSora-2 [59] adopts a $4 \times 32 \times 32$ VAE, and LTX [9] adopts an $8 \times 32 \times 32$ VAE to reduce the dimensionality of the latent features input to the DiT and results in faster generation speed. However, there has been limited research on how the VAE compression ratio affects the quality and speed of video generation. Upon aggressive compression, it becomes more challenging for the VAE decoder to fully reconstruct the details, which may result in quality loss. In this work, we perform a comprehensive study on the scaling of the VAE compression ratio. We follow [9, 48] and construct video VAEs with various compression ratios from $4 \times 16 \times 16$ to $8 \times 64 \times 64$. We build the VAE with 3D convolutions to better handle video modality, and use a fixed latent channel number for all variants. We train the same DiT network under each latent space and benchmark the generation speed and quality. Detailed results and discussions can be found in Sec. 5.3.

4.2 Efficient DiT Architecture via KD-Guided Tri-Level Pruning

Despite operating in a highly compressed latent space, the size of the Diffusion Transformer (DiT) remains a critical factor in edge generation scenarios [49], where mobile devices are constrained by

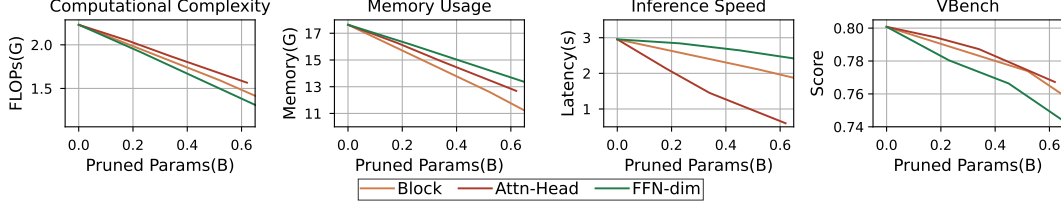


Figure 3: Sensitivity Analysis of DiT Components. The sensitivity analysis is conducted by progressively pruning DiT blocks, attention-heads and feed-forward network (FFN) dimension. For each setting, we benchmark FLOPs, memory usage, inference speed, and VBench score to assess the impact of each component on model efficiency and performance.

limited memory, power, and computational resources. Training a compact DiT that still achieves high-quality generation is a non-trivial challenge. First, DiT models generally exhibit strong generation capabilities only when scaled to a sufficiently large capacity. Moreover, designing an effective small-scale DiT is difficult due to the high-dimensional design space—including network depth (number of transformer blocks), width (channel size), and attention head count.

A promising alternative is to begin with a well-trained large model and prune it to meet resource constraints. Prior work such as TinyFusion [7] explored this approach via block-wise pruning using a learnable layer mask, effectively constructing a shallower DiT. However, as the properties demonstrated in Figure 3, it is still of great value to design a fine-grained pruning method that offers deeper insight into which parameters are critical or redundant, thereby enabling a better trade-off between efficiency and generation quality.

To address these, we propose a tri-level pruning scheme combined with knowledge alignment, enabling us to derive an efficient DiT architecture from a larger teacher model. Our approach maintains competitive performance while meeting the requirements for edge deployment.

4.2.1 Tri-level Pruning

As exhibited in Figure 3, the transformer block pruning is a simple yet coarse approach, we consider it a low-granularity pruning scheme. To enable finer granularity and better address redundancies, we propose a tri-level pruning scheme that incorporates block pruning and further introduces fine-grained pruning techniques, including head pruning for the multi-head attention mechanism and channel pruning for the linear layer. Notably, the pruned model can be converted into a dense and compact form, enabling execution on mobile devices without requiring additional compilation or specialized hardware support.

We employ a set of learnable binary masks to implement the tri-level pruning scheme. Each binary mask encodes the importance of its corresponding granularity, *i.e.* block, attention-head, or linear dimension. A mask value of 0 indicates that the corresponding unit should be pruned, while a value of 1 denotes that it should be preserved. Specifically, block pruning can be formulated as shown in Eq. (3), where y_{b_i}, x_{b_i} indicate the input and output features of the b_i^{th} DiT block, $m_{b_i} \in \{0, 1\}$ is the binary mask associated with that block, and $\mathcal{M}_b = [m_{b_1}, \dots, m_{b_N}] \in \{0, 1\}^N$ denotes the set of binary masks for all DiT blocks. When $m_{b_i} = 1$, the block is active; otherwise, its output is bypassed through a residual connection.

$$y_{b_i} = \text{Block}_{b_i}(x_{b_i}) \odot m_{b_i} + x_{b_i} \odot (1 - m_{b_i}), \quad m_{b_i} \in \{0, 1\}, \mathcal{M}_b = [m_{b_1}, \dots, m_{b_N}] \in \{0, 1\}^N \quad (3)$$

The other two pruning schemes can be expressed using a unified formulation, since pruning attention heads is equivalent to removing specific output features before the multi-head attention operation for each token. By integrating the pruning mechanism into the linear layer, the operation can be formulated as shown in Eq. (4):

$$y_{l_i} = \text{Linear}_i(x_{l_i}, W_{l_i}, b_{l_i}) \odot \mathcal{M}_{l_i}, \quad m_{l_i}^d \in \{0, 1\}, \mathcal{M}_{l_i} = [m_{l_i}^1, \dots, m_{l_i}^D] \in \{0, 1\}^D \quad (4)$$

where y_{l_i}, x_{l_i} denote the input and output features of the l_i^{th} linear layer, and $\mathcal{M}_{l_i} \in \{0, 1\}^D$ is a binary mask with D -dimension corresponding to the output channels. For each $m_{l_i}^d \in \mathcal{M}_{l_i}$, a value of $m_{l_i}^d = 0$ zeros out the corresponding output channel at dimension d for layer l_i^{th} ; otherwise the channel remains active.

The proposed tri-level pruning scheme begins by generating a candidate mask set \mathbb{M} for each pruning target, as illustrated in Figure 2. These candidate masks are selected based on the desired number of active components, which are constrained by the memory limitation of the target device. Notably, exhaustively exploring all pruning combinations results in an extremely large search space, making the optimization problem intractable (*e.g.*, pruning 6 out of 32 attention heads results in 906,192 possible configurations). To mitigate this, we adopt a group-wise masking mechanism that partitions overall search space into smaller subspaces, allowing pruning to be performed efficiently within each subspace. Once the candidate masks are generated, we further optimize them to identify the optimal configuration that minimizes the information loss caused by pruning.

Here, we specify the details of our tri-level pruning scheme for constructing an efficient Diffusion Transformer architecture tailored to the iPhone 16 Pro Max. Due to the memory limitation of the device, the total number of parameters must remain under 1 billion. Based on the sensitivity analysis in Figure 3, which indicates that FFN contributes more significantly to performance than attention heads, we prioritize pruning attention heads more aggressively. Starting from a 2B parameter base model with 28 DiT blocks, 32 attention-heads, and FFN dimension of 8192, our final efficient model archives 915M parameters by pruning 8 blocks, 12 attention heads, and reducing the FFN dimension by 25%. Detailed results of the pruned model can be found in Sec. 5.

4.2.2 Knowledge Distillation via Feature Alignment

Knowledge distillation (KD) is a widely adopted technique for transferring knowledge from a teacher model to a student model. Therefore, it is an effective strategy for preserving the performance of the pruned model. However, due to varying pruning schemes, the pruned student model may have different feature widths compared to the teacher model, which poses challenges for traditional distillation. Inspired by [56], we employ a trainable affine transformation to align the features between the teacher and the student model. Thus, distillation is then performed using the aligned features. This process is formally defined in Eq. (5), where y_{t_i} and y_{s_i} represent the output features of i^{th} DiT block group for the teacher model and the student model respectively:

$$\mathcal{L}_{\text{distill}} = \frac{1}{N} \sum_{i=1}^N \text{sim}(y_{t_i}, \mathcal{F}_i(y_{s_i}; \Theta_i)); \quad (5)$$

Here, N denotes the number of DiT block groups, and $\text{sim}(\cdot, \cdot)$ is a similarity alignment function used to match the feature distributions between teacher and student. The function $\mathcal{F}_i(\cdot; \Theta_i)$ is an affine transformation parameterized by Θ_i , introduced to align the dimensionality of the student features with that of the teacher. The overall training loss is formulated as Eq. (6), where $\mathcal{L}_{\text{flow-matching}}$ is the conditional flow-matching objective from Eq. (2), and α is a hyper-parameter to adjust the weight of distillation. α is set to 0.01 in our experiments.

$$\mathcal{L} = \mathcal{L}_{\text{flow-matching}} + \alpha \mathcal{L}_{\text{distill}} \quad (6)$$

4.3 Adversarial Finetuning for Step Distillation

In this section, we describe our step distillation procedure. Following earlier works [49], we design an adversarial distillation procedure involving a generator $\mathcal{G}_\theta(t, \mathbf{x}_t)$ and a discriminator $\mathcal{D}\phi(t, \mathbf{x}_t)$ network. We initialize the generator with pre-trained DiT denoiser weights (v_θ) from the diffusion training (see Eq. (2)). For the discriminator $\mathcal{D}\phi$, we create a similar DiT model, where the weights for the initial K DiT blocks are inherited from the generator \mathcal{G}_θ . We keep these blocks frozen during the discriminator training and append a prediction layer on top of the learnable DiT block to enable discrimination of fake generations and real video data. Note that this design allows the discriminator to be timestep-aware and be aligned well with the generator while it evolves during training. Additionally, we include additional learnable DiT blocks in discriminator architecture, the spatio-temporal representation capacity of the network is greatly enhanced due to the presence of full 3D self-attention and cross-attention layers. A simple multi-layer perception (MLP) with SiLU activation is appended after the DiT blocks to serve as the classification head.

To obtain a k -step distillation procedure, we predefine the intermediate diffusion timesteps as $\mathcal{T} = \{T_1, T_2, \dots, T_k\}$ with the following ordering $T_1 = 1 > T_2 > \dots > T_k > 0$. Typically, k is set to 4 to achieve a 4-step diffusion model. Given a real data sample \mathbf{X}_0 , we can obtain the latent \mathbf{x}_0 using the VAE. We sample two timesteps t and t' uniformly at random from the set \mathcal{T} such that

$t' < t$. We can construct the fake and real samples using the diffusion forward Eq. (1) and velocity from the generator $\mathcal{G}_\theta(t, \mathbf{x}_t)$ as follows:

$$\text{Fake : } \hat{\mathbf{x}}_{t'} = \mathbf{x}_t + (t' - t) \cdot \mathcal{G}_\theta(t, \mathbf{x}_t); \quad \text{Real : } \mathbf{x}_{t'} = (1 - t') \mathbf{x}_0 + t' \epsilon; \epsilon \sim \mathcal{N}(0, I) \quad (7)$$

Using the above real and fake samples, we can define the discriminator and generator losses. Below, we employ the widely used [38, 37, 36, 57] hinge loss [23] as the adversarial training objective. The discriminator’s goal is to differentiate between real and fake samples by minimizing:

$$\mathcal{L}_{\text{adv}}^{\mathcal{D}} = \mathbb{E}_{t', \mathbf{x}_0} [\text{ReLU}(1 + \mathcal{D}_\phi(\mathbf{x}_{t'}, t'))] + \mathbb{E}_{t, t', \mathbf{x}_0} [\text{ReLU}(1 - \mathcal{D}_\phi(\hat{\mathbf{x}}_{t'}, t'))], \quad (8)$$

The adversarial objective for the generator $\mathcal{L}_{\text{adv}}^{\mathcal{G}}$ and the reconstruction objective $\mathcal{L}_{\text{recon}}$ are defined as:

$$\mathcal{L}_{\text{adv}}^{\mathcal{G}} = \mathbb{E}_{t, t', \mathbf{x}_0} [\mathcal{D}_\phi(\hat{\mathbf{x}}_{t'}, t')]; \quad \mathcal{L}_{\text{recon}} = \sqrt{\|\hat{\mathbf{x}}_0 - \mathbf{x}_0\|_2^2 + c^2} - c. \quad (9)$$

where $\hat{\mathbf{x}}_0 = \mathbf{x}_t - t \cdot \mathcal{G}_\theta(t, \mathbf{x}_t)$, and $c > 0$ is an adjustable constant. Following [57, 14], we also incorporate a reconstruction objective to enhance training stability.

5 Experiments

Training. All training is done on internal collected image and video data, including both real and synthetic data. We use 128 NVIDIA A100 80GB GPUs for DiT training, using AdamW optimizer with $5e-5$ learning rate and betas values as $[0.9, 0.999]$. We build our Diffusion Transformer following public models [53, 9] using Diffusers library [43], and incorporate QK normalizations and Rotary Positional Embeddings (RoPE) [41]. To further ensure memory efficiency for mobile deployment, we adopt RMSNorm throughout the network.

Adversarial Fine-tuning is conducted for $20K$ iterations on 64 A100 GPUs, using the AdamW optimizer with a learning rate of $1e-6$ for the generator (*i.e.*, DiT) and $1e-4$ for the discriminator heads. We set the betas as $[0.9, 0.999]$ for the generator optimizer, and $[0.9, 0.999]$ for the discriminator optimizer. We set the EMA rate as 0.95 following SF-V [57]. We set $m = -1, s = 1$ for logit-norm if not otherwise noted.

Evaluation. Our model is evaluated following the standard VBench [15] setting, that is, we generate 5 videos for each prompt, and test the scores over the 1K prompt set. Using the adversarial distilled model, we generate 121-frame horizontal videos at a resolution of 576×1024 with 4 denoising steps, without classifier-free guidance. The generated video is saved at 5 seconds 24 FPS for score testing and qualitative visualization. We use different seeds and find the Δ VBench score is lower than ± 0.2 . For mobile demo, we generate $49 \times 384 \times 512$ videos on iPhone 16 Pro Max using CoremITools [1] under half precision. We measure the latency by 50 runs and take the median.

5.1 Qualitative Results

We show visualizations of our generated videos in Figure 4. Our model consistently produces high-quality video frames and smooth object movements. To demonstrate the generic text-to-video generation ability, we show various generation examples, including human, animal, photorealistic, and art-styled scenes. We include more video visualizations and the real-time model demo in *supplementary material*.

5.2 Quantitative Benchmark

We present a comprehensive evaluation of our method against existing popular video generation models on VBench [15], as in Tab. 2. Despite the fact that our model is compact and designated for fast inference on mobile, it achieves a higher total score compared to recent arts, including the DiT-based OpenSora-V1.2, CogVideoX-2B [53], and the UNet-based VideoCrafter-2.0 [4]. In addition, compared to the 4-step distilled T2V-Turbo [20] and AnimateLCM [46], our model achieves better performance with more than 50% reduction in size. The quantitative scores demonstrate the superiority of our efficient model design and the tailored adversarial distillation method. We include human evaluations of the generated videos in the *supplementary material*.



Figure 4: Video generated by our efficient diffusion transformer.

Model	Params (B)	Total	Quality	Semantic	Flickering	Aesthetics	Imaging	Obj. Class	Scene	Consistency
Wan2.1	14	84.70	85.64	80.95	99.53	61.53	67.28	94.24	53.67	27.44
Wan2.1	1.3	83.31	85.23	75.65	99.55	65.46	67.01	88.81	41.96	25.50
Open-Sora-2.0	11	84.34	85.40	80.12	99.40	64.39	65.66	94.50	52.71	27.50
Open-Sora V1.2	1.2	79.76	81.35	73.39	99.53	56.85	63.34	82.22	42.44	26.85
Hunyuan	13	83.24	85.09	75.82	99.44	60.36	67.56	86.10	53.88	26.44
CogVideoX1.5	5	82.01	82.72	79.17	98.53	62.07	65.34	83.42	53.28	27.42
CogVideoX	5	81.91	83.05	77.33	78.97	61.88	63.33	85.07	51.96	27.65
CogVideoX	2	81.55	82.48	77.81	98.85	61.07	62.37	86.48	50.04	27.33
Step-Video	30	81.83	84.46	71.28	99.40	61.23	70.63	80.56	24.38	27.12
Mochi-1	10	80.13	82.64	70.08	99.40	56.94	60.64	86.51	36.99	25.15
LTX-Video	1.8	80.00	82.30	70.79	99.34	59.81	60.28	83.45	51.07	25.19
Ours-Server	2.0	83.09	84.65	76.86	98.74	64.72	65.85	90.57	52.76	27.28
Ours-Mobile	0.9	81.45	83.12	74.76	98.11	64.16	63.41	92.26	51.06	25.51

Table 2: VBench [15] comparison with popular open-source Diffusion Transformer video generation models. Scores for open-source models are collected from the VBench Leaderboard.

5.3 Ablation Study

Scaling VAE Compression Ratio. In Tab. 3, we scale the VAE compression ratio and compare the DiT generation Speed and video quality. The generation speed is measured by testing one denoising step on the Nvidia A100 GPU with $121 \times 576 \times 1024$ resolution and the iPhone 16 PM with $49 \times 384 \times 512$ resolution. We observe that though low compression VAEs ($4 \times 16 \times 16$) can achieve better reconstruction PSNR, the generation speed is slower by magnitudes, and it encounters OOM on mobile device. On the other hand, aggressive compression ($8 \times 64 \times 64$) results in poor reconstruction, and will negatively impact generation quality (VBench scores). We find that ($8 \times 32 \times 32$) hits a balance between speed and quality, and employ this configuration for our Diffusion Transformer. The training details for the video VAE can be found in *supplementary material*.

Impact of KD-Guided Training. We compare models trained with and without the proposed KD-guided training to show its impact. As shown in Tab. 4, the tri-level pruned model trained with our loss function outperforms the version trained without it, particularly in terms of semantic scores.

VAE	PSNR	A100	iPhone	Total	Quality	Semantic	Aesthetic	Consistency	Flickering
$4 \times 16 \times 16$	33.1	7900	✗	80.35	82.05	73.54	64.45	26.80	98.59
$4 \times 32 \times 32$	30.9	920	3315	79.95	82.99	67.83	61.52	27.07	97.46
$8 \times 32 \times 32$	30.6	380	880	79.80	82.59	68.66	61.80	27.17	97.70
$8 \times 64 \times 64$	28.2	90	155	78.40	81.79	64.86	55.29	26.11	97.52

Table 3: Scaling VAE compression ratio. VAE PSNR is measured on DAVIS [30] with $33 \times 512 \times 512$ resolution. Latencies are reported in *ms* for one denoising step, and we test with $121 \times 576 \times 1024$ resolution for GPU and $49 \times 384 \times 512$ for iPhone. VBench scores are also provided for each variant.

KD-Guided Tri-Level Pruning. We compare our pruned mobile network with the direct training of small networks. We ablate two variants of compact network design, *i.e.*, a wide but shallow DiT with 2048 hidden dimensions and 14 blocks, and a narrow but deep DiT with 1472 hidden dimensions and 28 blocks. To ensure a fair comparison, all models are trained for 200K iterations, combined training cost of supernet pretraining and pruning. As shown in Tab. 4, our KD-Guided tri-level pruning consistently outperforms narrow or shallow variants that are directly trained from scratch. These results also aligned with the analysis in Figure 3, indicating that aggressively pruning along a single granularity fails to achieve optimal performance.

Method	KD	Params(M)	Quality	Semantic	Total
Tri-Level	✓	915	83.12	74.76	81.45
Tri-Level	✗	915	82.19	66.23	79.00
shallow	✗	932	81.52	67.01	78.62
narrow	✗	945	80.92	65.87	77.92

Table 4: Ablation study on tri-level pruning schemes and fine-tuning using proposed knowledge distillation.

Head	#Steps	Quality	Semantic	Total
DiT block + MLP	4	83.81	72.89	81.63
ResBlock-2D + Temporal-Attn [49]	4	83.24	67.78	80.14
Lightweight ResBlock [45]	4	80.05	66.01	77.24

Table 5: Ablation study on different discriminator head design. The evaluation is conducted with a 4-step generation without classifier-free guidance.

Full Guidance Adversarial Distillation. We validate the effectiveness of the proposed discriminator tailored for the DiT denoiser through an ablation study on its prediction head. The experiments are conducted using our pre-trained 2B parameter DiT model with various discriminator head designs. We compare our design with spatial-temporal heads introduced in [49] and the lightweight ResBlock head proposed in [45]. It is worth noting that the convolutional head in [45] was originally designed for the text-to-image model; for a fair comparison in our text-to-video setting, we extend it to a Conv3D variant in the ablation. We show that the proposed DiT block-based discriminator head, incorporating 3D Self-Attention, Cross-Attention, FFN, and an additional MLP classification head, achieves best 4-step generation performance, outperforming other variants, particularly in terms of semantic scores.

6 Conclusion, Limitations and Broader Impact

In this work, we present an efficient video generation framework that significantly accelerates Diffusion Transformers, making real-time synthesis feasible on mobile devices. By combining a high-compression VAE, latency- and sensitivity-aware pruning, and adversarial step distillation, we successfully deploy DiT video generator to iPhone and reduce inference to just four steps while maintaining high visual quality. Our pipeline achieves over 10 FPS generation speed (4 seconds to generate 49 frames) on an iPhone 16 Pro Max, demonstrating the practical real-time viability of DiT-based video generation on edge devices.

Despite these advances, our method has several limitations. First, the highly compressed latent space and DiT pruning lead to occasional degradations in fine-grained details, particularly in fast motion or complex texture scenes. Second, due to various practical constraints, most state-of-the-art video diffusion models (VDMs) used for comparison in this work, including our own, are trained on internally collected video datasets that cannot be fully disclosed or released. As a result, direct comparisons may not be entirely fair and reproducible. To mitigate this limitation, we include a reproduction of the LTX model trained on our dataset and report the results in the *supplementary material*. This work enables efficient video generation on mobile devices, but also carries the potential risk of misuse for generating fake or inappropriate content.

References

- [1] Coremltools. <https://coremltools.readme.io/docs>.
- [2] N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, T. Cai, P. Chattopadhyay, Y. Chen, Y. Cui, Y. Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.
- [3] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.
- [4] H. Chen, Y. Zhang, X. Cun, M. Xia, X. Wang, C. Weng, and Y. Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7310–7320, 2024.
- [5] T. Dao, T. H. Nguyen, T. Le, D. Vu, K. Nguyen, C. Pham, and A. Tran. Swiftbrush v2: Make your one-step diffusion model better than its teacher. In *European Conference on Computer Vision*, pages 176–192. Springer, 2025.
- [6] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.
- [7] G. Fang, K. Li, X. Ma, and X. Wang. Tinyfusion: Diffusion transformers learned shallow. *arXiv preprint arXiv:2412.01199*, 2024.
- [8] K. Frans, D. Hafner, S. Levine, and P. Abbeel. One step diffusion via shortcut models, 2024.
- [9] Y. HaCohen, N. Chiprut, B. Brazowski, D. Shalem, D. Moshe, E. Richardson, E. Levin, G. Shiran, N. Zabari, O. Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.
- [10] Y. HaCohen, N. Chiprut, B. Brazowski, D. Shalem, D. Moshe, E. Richardson, E. Levin, G. Shiran, N. Zabari, O. Gordon, P. Panet, S. Weissbuch, V. Kulikov, Y. Bitterman, Z. Melumian, and O. Bibi. Ltx-video: Realtime video latent diffusion, 2024.
- [11] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.
- [12] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [13] E. Hogeboom, J. Heek, and T. Salimans. simple diffusion: End-to-end diffusion for high resolution images. *arXiv preprint arXiv:2301.11093*, 2023.
- [14] D. Hu, J. Chen, X. Huang, H. Coskun, A. Sahni, A. Gupta, A. Goyal, D. Lahiri, R. Singh, Y. Idelbayev, J. Cao, Y. Li, K.-T. Cheng, S.-H. Chan, M. Gong, S. Tulyakov, A. Kag, Y. Xu, and J. Ren. Snapgen: Taming high-resolution text-to-image models for mobile devices with efficient architectures and training. *arXiv:2412.09619 [cs.CV]*, 2024.
- [15] Z. Huang, Y. He, J. Yu, F. Zhang, C. Si, Y. Jiang, Y. Zhang, T. Wu, Q. Jin, N. Chanpaisit, Y. Wang, X. Chen, L. Wang, D. Lin, Y. Qiao, and Z. Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- [16] B. Kim, Y.-G. Hsieh, M. Klein, M. Cuturi, J. C. Ye, B. Kawar, and J. Thornton. Simple reflow: Improved techniques for fast flow models. *ArXiv preprint*, abs/2410.07815, 2024.
- [17] B. Kim, K. Lee, I. Jeong, J. Cheon, Y. Lee, and S. Lee. On-device sora: Enabling training-free diffusion-based text-to-video generation for mobile devices, 2025.
- [18] W. Kong, Q. Tian, Z. Zhang, R. Min, Z. Dai, J. Zhou, J. Xiong, X. Li, B. Wu, J. Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.
- [19] Kuaishou. Kling. <https://kling.kuaishou.com/en>.
- [20] J. Li, W. Feng, T.-J. Fu, X. Wang, S. Basu, W. Chen, and W. Y. Wang. T2v-turbo: Breaking the quality bottleneck of video consistency model with mixed reward feedback. *ArXiv preprint*, abs/2405.18750, 2024.

- [21] Y. Li, H. Wang, Q. Jin, J. Hu, P. Chemerys, Y. Fu, Y. Wang, S. Tulyakov, and J. Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *arXiv preprint arXiv:2306.00980*, 2023.
- [22] Y. Li, H. Wang, Q. Jin, J. Hu, P. Chemerys, Y. Fu, Y. Wang, S. Tulyakov, and J. Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- [23] J. H. Lim and J. C. Ye. Geometric gan. *ArXiv preprint*, abs/1705.02894, 2017.
- [24] B. Lin, Y. Ge, X. Cheng, Z. Li, B. Zhu, S. Wang, X. He, Y. Ye, S. Yuan, L. Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024.
- [25] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.
- [26] G. Ma, H. Huang, K. Yan, L. Chen, N. Duan, S. Yin, C. Wan, R. Ming, X. Song, X. Chen, et al. Step-video-t2v technical report: The practice, challenges, and future of video foundation model. *arXiv preprint arXiv:2502.10248*, 2025.
- [27] K. Mei, M. Delbracio, H. Talebi, Z. Tu, V. M. Patel, and P. Milanfar. Codi: Conditional diffusion distillation for higher-fidelity and faster image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9048–9058, 2024.
- [28] W. Menapace, A. Siarohin, I. Skorokhodov, E. Deyneka, T.-S. Chen, A. Kag, Y. Fang, A. Stoliar, E. Ricci, J. Ren, et al. Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. *arXiv preprint arXiv:2402.14797*, 2024.
- [29] OpenAI. Video generation models as world simulators. <https://openai.com/index/video-generation-models-as-world-simulators/>.
- [30] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 724–732, 2016.
- [31] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.
- [32] A. Polyak, A. Zohar, A. Brown, A. Tjandra, A. Sinha, A. Lee, A. Vyas, B. Shi, C.-Y. Ma, C.-Y. Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.
- [33] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.
- [34] T. Salimans and J. Ho. Progressive distillation for fast sampling of diffusion models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.
- [35] A. Sauer, F. Boesel, T. Dockhorn, A. Blattmann, P. Esser, and R. Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. *ArXiv preprint*, abs/2403.12015, 2024.
- [36] A. Sauer, K. Chitta, J. Müller, and A. Geiger. Projected gans converge faster. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17480–17492, 2021.
- [37] A. Sauer, T. Karras, S. Laine, A. Geiger, and T. Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 30105–30118. PMLR, 2023.
- [38] A. Sauer, D. Lorenz, A. Blattmann, and R. Rombach. Adversarial diffusion distillation. *ArXiv preprint*, abs/2311.17042, 2023.

- [39] Y. Song and P. Dhariwal. Improved techniques for training consistency models. *ArXiv preprint*, abs/2310.14189, 2023.
- [40] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency models. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 32211–32252. PMLR, 2023.
- [41] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [42] G. Team. Mochi, 2024.
- [43] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, D. Nair, S. Paul, W. Berman, Y. Xu, S. Liu, and T. Wolf. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>, 2022.
- [44] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, J. Wang, J. Zhang, J. Zhou, J. Wang, J. Chen, K. Zhu, K. Zhao, K. Yan, L. Huang, M. Feng, N. Zhang, P. Li, P. Wu, R. Chu, R. Feng, S. Zhang, S. Sun, T. Fang, T. Wang, T. Gui, T. Weng, T. Shen, W. Lin, W. Wang, W. Wang, W. Zhou, W. Wang, W. Shen, W. Yu, X. Shi, X. Huang, X. Xu, Y. Kou, Y. Lv, Y. Li, Y. Liu, Y. Wang, Y. Zhang, Y. Huang, Y. Li, Y. Wu, Y. Liu, Y. Pan, Y. Zheng, Y. Hong, Y. Shi, Y. Feng, Z. Jiang, Z. Han, Z.-F. Wu, and Z. Liu. Wan: Open and advanced large-scale video generative models, 2025.
- [45] F.-Y. Wang, Z. Huang, A. W. Bergman, D. Shen, P. Gao, M. Lingelbach, K. Sun, W. Bian, G. Song, Y. Liu, et al. Phased consistency model. *arXiv preprint arXiv:2405.18407*, 2024.
- [46] F.-Y. Wang, Z. Huang, X. Shi, W. Bian, G. Song, Y. Liu, and H. Li. Animatelcm: Accelerating the animation of personalized diffusion models and adapters with decoupled consistency learning. *ArXiv preprint*, abs/2402.00769, 2024.
- [47] F.-Y. Wang, L. Yang, Z. Huang, M. Wang, and H. Li. Rectified diffusion: Straightness is not your need in rectified flow. *ArXiv preprint*, abs/2410.07303, 2024.
- [48] Y. Wu, Y. Li, I. Skorokhodov, A. Kag, W. Menapace, S. Girish, A. Siarohin, Y. Wang, and S. Tulyakov. H3ae: High compression, high speed, and high quality autoencoder for video diffusion models. *arXiv preprint arXiv:2504.10567*, 2025.
- [49] Y. Wu, Z. Zhang, Y. Li, Y. Xu, A. Kag, Y. Sui, H. Coskun, K. Ma, A. Lebedev, J. Hu, D. Metaxas, Y. Wang, S. Tulyakov, and J. Ren. Snapgen-v: Generating a five-second video within five seconds on a mobile device, 2024.
- [50] Y. Xu, Y. Zhao, Z. Xiao, and T. Hou. Ufogen: You forward once large scale text-to-image generation via diffusion gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8196–8206, 2024.
- [51] H. B. Yahia, D. Korzhnikov, I. Lelekas, A. Ghodrati, and A. Habibian. Mobile video diffusion, 2024.
- [52] L. Yang, Z. Zhang, Z. Zhang, X. Liu, M. Xu, W. Zhang, C. Meng, S. Ermon, and B. Cui. Consistency flow matching: Defining straight flows with velocity consistency. *ArXiv preprint*, abs/2407.02398, 2024.
- [53] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.
- [54] T. Yin, M. Gharbi, T. Park, R. Zhang, E. Shechtman, F. Durand, and W. T. Freeman. Improved distribution matching distillation for fast image synthesis. *ArXiv preprint*, abs/2405.14867, 2024.
- [55] T. Yin, M. Gharbi, R. Zhang, E. Shechtman, F. Durand, W. T. Freeman, and T. Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6613–6623, 2024.
- [56] S. Yu, S. Kwak, H. Jang, J. Jeong, J. Huang, J. Shin, and S. Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.
- [57] Z. Zhang, Y. Li, Y. Wu, Y. Xu, A. Kag, I. Skorokhodov, W. Menapace, A. Siarohin, J. Cao, D. Metaxas, et al. Sf-v: Single forward video generation model. *ArXiv preprint*, abs/2406.04324, 2024.

- [58] Y. Zhao, Y. Xu, Z. Xiao, H. Jia, and T. Hou. Mobilediffusion: Instant text-to-image generation on mobile devices, 2024.
- [59] Z. Zheng, X. Peng, T. Yang, C. Shen, S. Li, H. Liu, Y. Zhou, T. Li, and Y. You. Open-sora: Democratizing efficient video production for all, 2024.